Kathleen Sullivan (SBN 242261)
kathleensullivan@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100

Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
John M. Neukom (SBN 275887)
johnneukom@quinnemanuel.com.
QUINN EMANUEL URQUHART & SULLIVAN LLP
50 California Street, 22nd Floor
San Francisco, CA 94111
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

Mark Tung (SBN 245782)
marktung@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
555 Twin Dolphin Drive, 5th Floor
Redwood Shores, CA 94065
Telephone: (650) 801-5000
Facsimile: (650) 801-5100

Steven Cherny *(admission pro hac vice pending)*
steven.cherny@kirkland.com
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, New York 10022
Telephone: (212) 446-4800
Facsimile: (212) 446-4900

Adam R. Alper (SBN 196834)
adam.alper@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, California 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

Michael W. De Vries (SBN 211001)
michael.devries@kirkland.com
KIRKLAND & ELLIS LLP
333 South Hope Street
Los Angeles, California 90071
Telephone: (213) 680-8400
Facsimile: (213) 680-8500

Attorneys for Plaintiff Cisco Systems, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA, SAN JOSE DIVISION

CISCO SYSTEMS, INC.,

Plaintiff,

vs.

ARISTA NETWORKS, INC.,

Defendant.

CASE NO. 5:14-cv-5344-BLF

PLAINTIFF CISCO SYSTEMS, INC.'S OPENING CLAIM CONSTRUCTION BRIEF

Tech Tutorial: Jan. 29, 2016
Claim Construction Hearing: Feb. 5, 2016

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page

Cases

Statutes

I. INTRODUCTION

For each of the claim terms in dispute, as explained below, Cisco's proposed construction is consistent with the intrinsic and extrinsic evidence. Arista, on the other hand, often attempts to rewrite the claims to include new limitations, or offers constructions for terms or phrases that can be understood according to plain and ordinary meaning. In many cases, Arista's constructions are at least inconsistent with the intrinsic and extrinsic evidence, if not outright contradicted by the evidence. For the reasons stated below, Arista's proposed constructions should be rejected.

II. BACKGROUND

A. U.S. Patent No. 7,047,526 ("'526 Patent")

The '526 Patent relates to "command and interface control of Operating Administration and Monitoring (OAM) executable routines within software systems." ('526 Pat., Ex. 1 at 1:7-9) These executable routines can be used in unified messaging or network management systems, for example. (*Id.* at 1:10-27). The '526 Patent describes the use of a parser that interprets user-entered text commands, which can be used to configure or manage, for example, network equipment such as routers and/or switches. The '526 Patent provides the user a set of universal commands, which are interpreted by the parser to determine the right management program to execute with the correct corresponding command. The patent refers to these universal commands as "generic commands." The parser uses a command parse tree, which contains elements of the generic commands, to determine a best match for the command entered by the user. The parser then issues a prescribed command for the management programs.

B. U.S. Patent No. 7,953,886 ("'886 Patent")

The '886 Patent relates to "routing systems for computer networks, and more particularly to the transmission of instructions to and receipt of data from such routing systems." ('886 Pat., Ex. 2 at 1:8-10). The '886 Patent explains that for routers using internetwork operating systems, access was generally accomplished via human input at the "command line interface" or "CLI." (*Id*. at 1:14-16). The '886 Patent discloses a system and method to allow for a more structured approach for accessing routers. (*Id*. at 1:34-35). To accomplish this, in one embodiment, CLI

command input statements and output statements are formatted in accordance with an XML schema of CLI rules and behaviors. (*Id*. at 3:22-26; *see also* Almeroth Decl. at ¶¶ 33-35).

III. LEGAL STANDARD

A. General Claim Construction Principles

Claim construction is a question of law, resolved by the court. *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 387 (1996). Claim construction "is not an obligatory exercise in redundancy," and is used only "to clarify and when necessary to explain what the patentee covered by the claims." *O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co*., 521 F.3d 1351, 1362 (Fed. Cir. 2008). When construing claims, it is improper to adopt constructions that appear "nowhere in the patent claims, specification, or prosecution history." *Outside Box Innovations, LLC v. Travel Caddy, Inc.*, 260 F. App'x 316, 319 (Fed. Cir. 2008).

Claim construction begins with the language of the claim itself. *See, e.g., Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005). Claim language should generally be given its plain and ordinary meaning, as there is a "'heavy presumption' that claim terms carry their accustomed meaning." *Home Diagnostics, Inc. v. LifeScan, Inc.*, 381 F.3d 1352, 1355 (Fed. Cir. 2004). In these cases, it is "proper for the district court not to construe [these terms]." *Silicon Graphics, Inc. v. ATI Tech's., Inc.,* 607 F.3d 784, 798 (Fed. Cir. 2010). Beyond the claims, the specification "is the single best guide to the meaning of a disputed term." *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996).

A patent's prosecution history may provide additional guidance in construing its claims. However, the prosecution history "is less useful for claim construction purposes." *Phillips*, 415 F.3d at 1317. And finally, though it will always be less significant than the intrinsic evidence, extrinsic evidence such as dictionaries, expert declarations, and learned treatises may provide additional context for a claim "if the court deems it helpful in determining 'the true meaning of language used in the patent claims.'" *Phillips*, 415 F.3d at 1318.

B. Construction of Means-Plus-Function Claims

Under 35 U.S.C. § 112 (f), a patent claim may be expressed as a "means" for performing a function without specifying the structure which performs that function. "Claim construction of a

means-plus-function limitation includes two steps. First, the court must determine the claimed function. Second, the court must identify the corresponding structure in the written description of the patent that performs that function. " *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 448 F.3d 1324, 1332 (Fed. Cir. 2006) (internal citations omitted); *see also* 35 U.S.C. § 112 (f).

IV. CISCO'S '526 PATENT

A. "management programs" in Claims 1, 10, 14, and 23

Cisco's proposed construction	Arista's proposed construction
"separate tools or external agents having their own respective command formats that provide management functions"	"tools that are configured to execute user-entered commands having their own respective command formats rather than the generic command format"

Cisco proposes a construction is fully consistent with the intrinsic record, while Arista's construction imposes requirements not supported by the intrinsic record and excludes important characteristics such as that they must be separate or external.

The parties' proposed constructions differ on several important points. First, Cisco proposes that the management programs must be "separate tools" or "external agents." A management program is not just a different part of the same module or routine that is executed – it must be separate. The specification describes the management programs as "external agents," "external programs," and "different tools." *See, e.g.,* :

> As shown in FIG. **1**, the management programs **18**, implemented for example by ***different*** OAM ***tools*** such as RTM programs, may be executed within the processor based system or externally as ***external agents*** accessible using a prescribed application programming interface (API). The management programs **18** may provide different administration and maintenance functions, for example initiating various real-time screens used to monitor the internal state of executable processes within the software based system **10**; alternately, ***different tools*** **18** may allow the user to control the various states within the various component of the software based system **10** via ***external programs*** (e.g., programs **18** *c* or **18** *d*), or may be used to issue ***external alarms*** (e.g., SNMP manager scripts) for ***external routines*** such as message waiting indicator routines.

('526 Pat., Ex. 1 at 3:1-15 (emphasis added); Almeroth Decl. at ¶¶ 59-61). The description of management programs clearly contemplates the use of external programs or agents, or alternatively, if the tools are executed, the use of separate and different tools or programs.

Second, Arista proposes that the management programs must have their own respective formats "rather than" the generic command format. Arista's use of "rather than" suggests that the command formats for the management programs must be different than the generic commands, but nothing in the intrinsic record requires this. The purpose of the generic commands is to allow the user not to have to remember the exact format and syntax of the commands for each management program (*id.* at 1:41-47), and can be achieved regardless if there is overlap between the generic commands and the management programs command format or not. Indeed, in Appendix Part A of the '526 Patent, some generic commands have substantial overlap with the corresponding management program command (*e.g., compare* "reload sched all" with "reloadsched"). The abstract describes that the generic commands are "independent from" the management program command formats, not "different from." Independence suggests there may or may not be overlap – the choices of generic commands and management program commands are simply made independently, without regard to the amount of overlap.

Third, Arista's proposal requires that the management programs are "configured to execute user-entered commands." There is nothing in the '526 Patent that requires the management program to directly execute user-entered commands. Instead, the '526 Patent contemplates a user entering a generic command, which a parser interprets so that the appropriate command can be issued to the management program. (*See, e.g.,* '526 Pat., Ex. 1 at cl. 1, 10, 14, 23). Arista's proposed construction therefore should be rejected.

B. "generic command" in Claims 1, 6, 10, 13, 14, 15, 19, and 23

Cisco's proposed construction	Arista's proposed construction
"command that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue the command based on the relative functions, as opposed to the specific syntax for a corresponding tool"	"command having a format and syntax that is an abstraction of the command formats and syntaxes of more than one management program, as opposed to the specific syntax for any such management program"

Cisco's construction is based very precisely on the description of generic command in the '526 Patent specification, while Arista's construction incorporates unnecessary and unsupported differences. (Almeroth Decl. at ¶¶ 62-63). The most significant difference is that Arista's

construction requires that the generic command be an abstraction of "more than one management program[,]" a limitation not supported by the intrinsic record (Almeroth Decl. at ¶¶ 65-67).

Cisco's construction is word-for-word based on the specification: "As illustrated in Part A of the attached appendix, the new syntax provides a generic instruction set that ***provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue [the] command based on the relative functions, as opposed to the specific syntax for a corresponding tool*** 18." ('526 Pat. at 3:31-35 (emphasis added); Almeroth Decl. at ¶ 68). Arista's construction adds in the extraneous concept that the generic command must be an abstraction of more than one management program, which is not supported by the specification. (Almeroth Decl. at ¶¶ 65-69). For example, in Appendix Part A, there are many generic commands which are mapped to only a single command for the management programs. (*See, e.g.,* '526 Pat., Ex. 1 at Appendix Part A ("stop system" maps to "obs –o APP –s down")). There are also generic commands that map to more than one management program. (*See, e.g., id.* ("watch acb globals" maps to both "BASEview" and "APPview")). Both possibilities exist, indicating a generic command is not defined by the number of management programs or management program commands it maps to.

Arista's construction also presents the following problem. Suppose a system provides a set of commands for the user so that the user does not need to learn the specific syntax of a corresponding tool. If there is only a single management program in this system because additional management programs have not yet been installed, do the user entered commands fail to be "generic commands" until additional management programs are installed? What if the system already had code that contained the mapping between the user-entered commands and additional management programs? It would be a strange result that the same user-entered command, in the same system, would be a "generic command" (or not) based on the state of installation of other programs in the system. (Almeroth Decl. at ¶¶ 64 and 67).

Arista may be trying to propose the idea that one cannot abstract the commands of only one single management program. But this misunderstands what an abstraction is. An abstraction is "the act or process of leaving out of consideration one or more qualities of a complex object so as to attend to others." (Ex. 8, Webster's Third New Int'l. Dictionary (2002); Almeroth Decl. at

¶¶ 66-70). An abstraction is not defined by how many ideas it captures (or how many underlying programs are included); rather, an abstraction means the user does not need to know the details of how something is implemented. Moreover, since the idea of abstraction is already expressly captured in Cisco's proposed construction, there is no reason to include "more than one management program" in the construction.

Arista's construction also unjustifiably embeds the idea of management program into the construction of "generic command," when such a connection is not part of the meaning of the term. A "generic command" does not need to be an abstraction of a management program – it could be the abstraction of any tool, whether it's used for management functions or not. The relationship between "generic command" and "management programs" in the claim language should not and does not change the meaning of "generic command."

Lastly, Arista's proposed construction is ambiguous when it refers to "the command formats and syntaxes of more than one management program." Because Arista's construction requires the command formats and syntaxes to be of "more than one" management program, the scope of what constitutes a "command format and syntax" of a single management program is important, and Arista's construction does not make that clear. Do all the commands of a management program share the same format and syntax? Does each different command of a management program necessarily have a different format and syntax by virtue of consisting of different words? Can different management programs share the same formats and syntaxes? Arista's construction would introduce ambiguity that does not help clarify the scope of the claim term.

C. "respective command formats," in Claim 1, 10, 14, 23

Cisco's Construction	Arista's Construction
"command format specific to a management program"	command names and syntaxes specific to each management program

The parties' dispute boils down to two issues: (1) whether the term "command format" should be given its plain and ordinary meaning, as Cisco contends, or should be limited to

"command names and syntaxes," as Arista contends, and (2) whether "respective" refers to "a management program," as Cisco contends, or "each management program," as Arista contends.

First, the Court need not construe "command format" because it is used according to its plain and ordinary meaning. *Home Diagnostics,* 381 F.3d at 1355. Arista's construction is improper for several reasons. First, Arista's construction improperly imports a phrase, "names and syntaxes," that appears in the '526 Patent specification's description of the related art, but never in the description of the invention itself. ('526 Pat., Ex. 1 at 1:31-37). Thus, the patentee was well aware of the phrase, but chose not to use it in the claim language or the specification in connection with the invention. Even if the phrase were used to describe an embodiment of the invention of the '526 Patent, which it is not, it is well settled that the Court should not import limitations from the specification. *Enercon GmbH v. International Trade Com'n*, 151 F.3d 1376, 1384 (Fed. Cir. 1998). Next, Arista's use of "names and syntaxes" introduces ambiguity by using two terms whose meanings, and therefore scope, may not be the same. Lastly, Arista's construction blurs together two concepts the '526 Patent keeps distinct: formats and syntaxes. One of the expressly described benefits of the invention of the '526 Patent is to eliminate the need for a user to "learn[] detailed ***command formats and syntax***." (*See, e.g., id*. at 1:52-62 (emphasis added); *see also id.* at Abstract (same)). Arista's construction equates "format" with "names and syntaxes," making the phrase "command formats and syntax" in the above passage redundant.

Separately, Arista's proposed construction imposes requirements not found in the claims or the specification. The claim language and specification state that management programs have respective command syntax, (*see e.g., id*. at 4:55-58 ("multiple [management programs] that have respective command syntax")), but this does not require, as Arista's construction implies, that ***each*** management program has a ***unique*** respective command syntax. The intrinsic record provides no support for Arista's "each" requirement, which should be rejected.

D. "command parse tree," in Claims 1, 3, 10, 11, 12, 14-16

Cisco's Construction	Arista's Construction
"a hierarchical data	"tree": "data structure consisting of linked nodes, with a root node (a

representation having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value"	node with no parent nodes), and where the remaining nodes are either a branch node (a node with a parent node and one or more children nodes), or a leaf node (a node with a parent node and no children nodes)" "command parse tree": "tree for interpreting commands where each node corresponds to a command component"

Cisco proposes adopting the plain and ordinary meaning of the term, while Arista's proposed construction confuses, rather than clarifies, the meaning of the term, and should be rejected. *See Stanacard, LLC v. Rebtel Networks,* 680 F. Supp. 2d 483, 493 (S.D.N.Y. 2010). The parties' dispute centers on two issues: (1) whether a "tree" must have more than one element/node[1] and (2) whether each and every element of the tree "corresponds to a command component."

Cisco's construction relies upon the plain and ordinary meaning of the term "tree," is based directly on claim language and specification, and does not alter the scope of the claim term. Arista's proposed construction for the word "tree" runs afoul of its own proffered extrinsic evidence by requiring at least two nodes. In particular, the Microsoft dictionary that Arista relies on for the meaning of "tree" does not require at least two nodes. (Dkt. 70-2 at 3 (quoting *Microsoft Press Computer Dictionary,* 3d ed. (1997) ("tree"))). Cisco is amenable to the interpretation of the word tree in this reference.

Second, Cisco's construction is taken directly from the claim language and specification. The claims and specification clearly describe a command parse tree as "having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value." ('526 Pat., Ex. 1 at Abstract, 1:51-54; 2:4-6; 2:-20-24; cl. 1, 10, 14, and 23). There is no reason to stray from this clear and unambiguous guidance.

In contrast, Arista's construction does little to clarify the meaning of the disputed term for two reasons. First, Arista's construction requires that the command parse tree be "a tree for interpreting commands[,]" a phrase that is *less* clear than the claim term because it introduces an

[1] Although Cisco's construction uses the word "element," as found in the '526 Patent, and Arista's construction uses "node," it is not apparent there is a substantive dispute on the difference between these terms. Cisco proposes "element" because it is the terminology used in the '526 Patent.

unnecessary additional term "interpreting." *Stanacard,* 680 F. Supp. 2d at 493. "[I]nterpreting" focuses on what a command parse tree is used for, which is improper. Second, Arista's construction requires that "each node corresponds to a command component[.]" There is no intrinsic support for a one-to-one correspondence between each node and a single command component. Instead, throughout the specification and the claims, the elements are described as "specifying ***at least one*** corresponding generic command component." (*See, e.g.,* '526 Pat., Ex. 1 at Abstract; 1:51-54; 2:1-8; 2:15-24; Claims 1, 10, 14, 23).

E. "the validating step including identifying one of the elements as a best match relative to the generic command" in Claims 1 and 14

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	Indefinite, OR if not indefinite: "the validating step having the capability of both identifying the element in the parse tree that exactly matches the generic command, and, in the absence of an exact match, identifying the element that contains the last validated component of the generic command"

The Court need not construe this term because it is used according to its plain and ordinary meaning. *See, e.g., Home Diagnostics,* 381 F.3d at 1355. Arista contends the term is indefinite or should have a specific meaning that differs from the plain and ordinary meaning.[2]

The Court need not construe this term. The specification provides an express example of this claimed verification step. ('526 Pat., Ex. 1 at 3:56-61). In one embodiment, this validating step is carried out by "the parser 14, [which] recursively traverses the command parse tree 22 for each command word to identify the best match for the generic command." (*Id.*).

In comparison, Arista's construction introduces limitations unsupported by the intrinsic record in certain respects and interjects unnecessary ambiguity in another respect. First, Arista's construction requires that the validating step be capable of "an exact match." The '526 Patent does not disclose any embodiments with Arista's "exact match" requirement, but rather, the '526 Patent discloses that the validating step based on a "best match" consists of matching a token

[2] Because Arista bears the burden on proving that a term is indefinite, Cisco reserves the right to respond to any such arguments raised in Arista's responsive brief.

based on "a portion of the generic command" of the token. (*Id*. at 3:56-61). There is simply no requirement in the specification, or otherwise, that the "best match" validation step must include the capability to determine an "exact match."

Moreover, Arista's contrived construction adds ambiguity into the meaning of the term by introducing a phrase that is foreign to the intrinsic record—"last validated component." First, the term at issue is a validating step of a "generic command," not a command component. Second, "last" is ambiguous. For example, it could refer to the generic command word appearing at the end of the user-entered generic command, or the last in time validated component.

Lastly, Arista's proposed construction of "the validating step" requires it to "hav[e] the capability of both identifying … and …" It does not make sense to construe a "validating step" as having a capability, much less two capabilities. *Stanacard,* 680 F. Supp. 2d at 493. Thus, the Court should reject Arista's proposed construction.

F. "recursively traversing the command parse tree based on an order of the input command words" in Claims 3 and 16

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	"recursively": "by using a function that calls itself," "traversing the command parse tree based on an order of the input command words": "sequentially determining the presence of each word of an input command in a node of a command parse tree, such that the order of the words (e.g., first, second, third) corresponds to the hierarchy of the nodes (e.g., parent, child, grandchild)."

Cisco contends that the Court should apply the "heavy presumption" that the plain and ordinary meaning applies and no construction is necessary. *Home Diagnostics,* 381 F.3d at 1355; Almeroth Decl. at ¶ 70. Arista's construction requires reading certain terms out of context and imposes restrictions not contemplated by the '526 Patent. (Almeroth Decl. at ¶¶ 71, 78-79).

Arista's construction exemplifies the precise reason that the Court should adopt the plain and ordinary meaning of the term. The '526 Patent claims recite (and the specification describes) "[r]ecursively traversing the command parse tree based on an order of the input command words." A tree is a hierarchical structure comprised of sub-trees that themselves can be treated as trees.

The claims require traversing the command parse tree in a recursive manner and is understood in the art as processing all nodes of a tree by systematically traversing each sub-tree of that tree until all sub-trees have been completed, and traversing each sub-tree using the same procedure (by traversing each sub-tree of the sub-tree). (Almeroth Decl. at ¶¶ 72-77).

Neither the claims nor the specification require implementing this traversal using a specific sequence of computer code, as Arista's construction requires. Indeed, the term "recursive" only appears in the '526 Patent when it is modifying the method of traversal. ('526 Pat., Ex. 1 at 3:55; cl. 3, 12, 16, and 25). The '526 Patent does not mandate that the computer code must use a specific type of "recursion" or recursive functions. (Almeroth Decl. at ¶ 79). Thus, it would be improper for the Court to limit the *manner* of traversal (the path of travel) to a specific *implementation* of that traversal (*e.g.* use of a recursive function). *See Silicon Graphics, Inc. v. ATI Technologies, Inc.*, 607 F.3d 784, 790-91 (Fed. Cir. 2010); Almeroth Decl. at ¶ 79.

The other portions of Arista's construction are likewise problematic as they lack any intrinsic or extrinsic support, and are contrary to the plain language of the claims. Arista's construction requires "determining the presence of each word of an input command[,]" contrary to claim 16's requirement of determining a "best match relative to the generic command." ('526 Pat., Ex. 1 at cl. 14). The specification describes best match as relying on "a ***portion*** of the generic command[,]" (*id*. at 3:55-56 (emphasis added)), instead of an exact match of every word in the generic command, as Arista's construction requires. Thus, the Court should reject Arista's attempt to stray from the plain meaning of the term.

G. "the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value" in Claims 1 and 14

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	"elements": "nodes" "command action value": "piece of data that uniquely represents the prescribed command." the entire phrase: "the command parse tree having nodes, such that each node specifies a unique command action value for each generic command

	component."

Cisco contends that this term should be given its plain and ordinary meaning, while Arista proposes a construction for the entire phrase, as well as two sub-parts—"nodes" and "command action value." As noted in fn. 1 above, it is not clear that there is a substantive disagreement between "element" and "node." The parties' dispute focuses on (1) what is meant by "data" and (2) whether the command action value must be "unique" to each generic command component.

It is not clear what is encompassed by "data" in Arista's construction as "data" is not found in the claims or the specification. Thus, at a minimum, there is no intrinsic support for the inclusion of "data" in Arista's construction. Furthermore, if by the use of "data," Arista means to limit command action value to only certain types such as numbers, symbols or characters, then such limitation is unjustified. The command action value may be implemented as a stored function (executable source code); series of numbers, symbols, and characters; or even a memory address, so long as it is "specified" by at least one element of the command parse tree. (*See, e.g.,* '526 Pat., Ex. 1 at cl. 1). Thus the Court should reject any attempt by Arista to unreasonably restrict the scope of command action value.

Likewise, Arista's inclusion of "unique" is contrary to the disclosures of the '526 Patent. For example, Appendix Part A discloses two, non-unique, command action values (corresponding to "BASEview" and "APPview") for the same generic command component ("watch acb globals"). Arista's construction would read out this express example and is thus, improper.

H. "prescribed command of a selected one of the management programs"[3] in Claims 1 and 14

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	"one of the commands in the respective command format of a specific management program"

[3] The disputed term appearing in the Joint Claim Construction and Prehearing Statement (Dkt. 70) is "issuing a prescribed command of a selected one of the management programs according to the corresponding command format, based on the identified one element", but the parties' only dispute is with respect to "prescribed command of a selected one of the management programs," as the parties agree the rest of the term does not need to be construed.

Cisco contends that the Court should apply the "heavy presumption" that the plain and ordinary meaning applies to the claim term and no construction is necessary. *Home Diagnostics,* 381 F.3d at 1355. Arista contends that the term should be construed.

Arista's proposed construction improperly requires that the prescribed command be in a command format that is specific to one, and only one, management program, a requirement that is not found in the specification or the prosecution history of the '526 Patent, and thus is improper. *Acumed LLC v. Stryker Corp.,* 483 F.3d 800, 807 (Fed. Cir. 2007). In addition, one of the purposes of the invention of the '526 Patent is that the user be allowed to use generic commands to control multiple management programs. ('526 Pat., Ex. 1 at 1:45-47). Arista's construction, by limiting the prescribed command to only a single, specific management program would defeat this purpose of the invention. Thus, the Court should reject Arista's construction and decline to construe this term.

I. "command word translation table, configured for storing for each prescribed command word a corresponding token" in Claim 2 and 15

Cisco's Construction	Arista's Construction
Plain and ordinary meaning (except for specific terms appearing within the phrase that are otherwise terms for construction)	"command word translation table": "data structure with rows and columns that translates a command word into a token" "prescribed command word": "valid generic command word." "token": "letter, number, or symbol that either (1) uniquely represents a prescribed command word, or (2) indicates the presence of an invalid command word" The remainder of this phrase does not require construction.

Cisco contends that the Court should apply the "heavy presumption" that the plain and ordinary meaning applies, while Arista asks the Court to construe three sub-parts of this phrase.

First, the term "command word translation table" is expressly defined in the claims as "configured for storing for each prescribed command word a corresponding token" and later to perform the validation step discussed above in Section IV.E. (*See, e.g.,* '526 Pat., Ex. 1 at 3:61). Thus, there is no need to provide additional definition of this term.

Arista's construction narrowly focuses on only one possible structure (data structure with rows and columns in Figure 2) for implementing the translation table, that is not even disclosed, much less mandated, by the '526 Patent specification. Figure 2 is described as "illustrating … an embodiment of the present invention." ('526 Pat., Ex. 1 at 2:48-49). As courts recognize, such figures, which are illustrative representations of an embodiment, should not narrowly restrict claim scope. *See Varco, L.P. v. Pason Sys. USA Corp.*, 436 F.3d 1368, 1375 (Fed. Cir. 2006).

Arista's construction is also improper as it adds ambiguity. *Stanacard,* 680 F. Supp. 2d at 493. Arista's construction of "prescribed command word" is not supported by the intrinsic record, which does not define, mention, or require that it be a "valid generic command word." Likewise, Arista's construction for "token" limiting it to a "letter, number, or symbol" is unnecessarily narrow in scope, improperly limiting the term to a single, illustrative embodiment, contrary to well-established claim construction principles. *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1326 (Fed. Cir. 2002).

Furthermore, Arista's construction is unsupported by the intrinsic record and should be rejected. For example, nothing in the claims or specification ***requires*** that a token must be a "letter, number, or symbol." Indeed, Arista's proposed construction would exclude a token comprised of more than one letter, number, or symbol, or any computer implementation other than a letter, number, or symbol. There is simply no basis for doing so and the Court should reject Arista's proposed construction.

J. "means for validating a generic command received from a user, the validating means configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating means identifying one of the elements as a best match relative to the generic command" in Claim 23

Cisco's Construction	Arista's Construction
For "means for validating a generic command received from a user": Function: validating a generic command received from a user. Structure: Parser 14 in Figure 2, which includes the command	Functions: (1) validating a generic command received from a user (2) specifying valid generic commands relative to a prescribed generic command format, (3) having elements each specifying at least one corresponding generic component and a corresponding at least one command action

word translation table 20 and the command parse tree 22, as described in 3:36- 61, and equivalents For rest of term, plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)	value, and (4) identifying one of the elements as a best match relative to the generic command. <u>Disclosed structure:</u> A processor executing a parser, and a corresponding memory storing a command parse tree, wherein the parser executes the algorithm of Figure 3, and wherein (1) each node of the command parse tree specifies one token and a corresponding command key; (2) the top-level nodes of the command parse tree represent all possible valid first words in the input command, second-level nodes represent all possible valid second words for each valid first word in the input command, and so on;

The parties' main dispute is two-fold: (1) the proper scope of the means plus function limitation and (2) the proposed structure. First, the Court should only construe the clause "means for validating a generic command received from a user." The remainder of the term merely adds additional sub-functions to the one means plus function clause as evident by the comma and clause that immediately follow: "***the validating means*** configured for[.]" ('526 Pat., Ex. 1 at cl. 23).

Second, Arista's proposed structure is improper because it impermissibly includes more than the structure necessary to perform the means plus function. *Wenger Mfg., Inc. v. Coating Machinery Systems, Inc.*, 239 F.3d 1225, 1233 (Fed. Cir. 2001) (legal error to import "structural limitations from the written description that are unnecessary to perform the claimed [mean plus] function."). Arista seeks to import the structure from the most restrictive embodiment, ('526 Pat., Ex. 1 at 4:17 ("For example…")), ignoring the less restrictive embodiment relied on by Cisco that also performs the claimed function. Worse yet, paragraph (2) of Arista's proposed structure has no basis in the specification. Neither the claims nor the specification require that the top-level nodes of the command parse tree represent all possible valid first words in the input command, second-level nodes represent all possible valid second words for each valid first word in the input command, and so on, as Arista contends.

In contrast, Cisco cites the structure for the broader description of the validating process, as shown in Figure 2 and described in the specification. According to the '526 Patent, Figure 2 discloses "in detail the parser…[which] includes a command word translation table 20 and a

command parse tree 22…[a] is configured for ***validating a received generic command by comparing each input command word*** to the command parse tree 22 to determine for the received generic command a tree element 24 identified as a best match." ('526 Pat., Ex. 1 at 3:36-51 (emphasis added)). This portion of "[t]he specification…clearly links or associates [these] structure[s] to the [validating function] recited in the claim," and thus the Court should adopt Cisco's proposed structure. *Omega Eng'g, Inc, v. Raytek Corp.*, 334 F.3d 1314, 1321 (Fed. Cir. 2003). The Court should adopt this concise, yet complete, description of the structure necessary for carrying out the validation function.

V. CISCO'S '886 PATENT

A. "extensible markup language (XML)" in Claims 1-10

Cisco's Construction	Arista's Construction
"extensible": a property of a computer language that allows the user to add new features or modify existing ones "markup language": a computer language that allows the user to add identifiers to a document for indicating logical components or layout	markup language defined by one of the versions of the XML standard published by the W3C organization

It is well-established law that "claims are 'of primary importance, in the effort to ascertain precisely what it is that is patented'" and that "the words of a claim 'are generally given their ordinary and customary meaning.'" Phillips, 415 F.3d at 1312-13. "Claim terms are entitled to a 'heavy presumption' that they carry their ordinary and customary meaning to those skilled in the art in light of the claim term's usage in the patent specification." *Elbex Video, Ltd. v. Sensormatic Elecs. Corp.*, 508 F.3d 1366, 1371 (Fed. Cir. 2007). Technical dictionaries are particularly useful in understanding the ordinary and customary meaning of claims terms. *Phillips*, 415 F.3d at 1318. In cases where "the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess," "the inventor's lexicography governs." *Id.* at 1316 (citation omitted). Simply put, Cisco's proposed construction comports with these well-established rules, while Arista's proposed construction does not.

All asserted claims of the '886 patent recite "an extensible markup language (XML)

format." As it is apparent in the context of the claims, "(XML)" is used as a shorthand abbreviation for "extensible markup language." These words—"extensible," "markup," and "language"—are well-known terms of art. As explained by Cisco's expert Dr. Almeroth, these terms are generic and do not denote to one of ordinary skill in the art a particular language. And in the 2003-2005 time frame, many different languages can be characterized as an "extensible markup language." (Almeroth Decl. at ¶¶ 39-43). Furthermore, neither the specification nor the file history reveals that the patentees applied special definitions to "extensible," "markup," or "language" that are different from their ordinary and customary meaning. Accordingly, the ordinary and customary meaning of these terms should apply, and the Court should help the jury understand the terms by supplying constructions that will define them terms in the context of the art. *Power-One, Inc. v. Artesyn Techs., Inc.*, 599 F.3d 1343, 1348 (Fed. Cir. 2010) ("The terms, as construed by the court, must 'ensure that the jury fully understands the court's claim construction rulings and what the patentee covered by the claims.'") (citation omitted). Cisco offers as its proposed construction the ordinary and customary definitions of the terms that are almost copied verbatim from technical dictionaries. For example, Cisco's proposed construction for "extensible" is taken from the dictionary definition for "extensible language":

> Cisco's proposed construction: "extensible": a property of a computer language that allows the user to ***add new features or modify existing ones***.
>
> McGraw-Hill Dictionary of Scientific and Technical Terms, 6th Ed (2003) (Ex. 6 at CSI-CLI-00403881): Extensible language: a programming language which can be modified by ***adding new features or changing existing ones***.

(emphasis added). Likewise, Cisco's proposed construction for "markup language" is taken from dictionary definitions of "markup" and "markup language":

> Cisco's proposed construction: "markup language": a computer language that allows the user to ***add identifiers to a document for indicating logical components or layout***
>
> McGraw-Hill Dictionary of Scientific and Technical Terms, 6th Ed (2003) (Ex. 6 at CSI-CLI-00403882): Markup: the process of ***adding information (tags) to an electronic document*** that are not part of the content but describe its structure or elements.

> Dictionary Of Computer Science, Engineering, and Technology (2001) (Ex. 7 at CSI-CLI-00403896): Markup language: one of any languages for annotation of source code to simply improve the source code's appearance with the means of bold-faced key words, slanted comments, etc. In computerized document preparation, ***a method of adding information to the text indicating the logical components of a document, or instructions for layout of the text on the page*** or other information which can be interpreted by some automatic system.

(emphasis added). Accordingly, Cisco's proposed construction is consistent with the ordinary and customary meaning of the terms and supplies definitions that will help the jury understand the claim language. (Almeroth Decl. at ¶ 44).

Arista, on the other hand, attempts to improperly limit the asserted claims to a particular extensible markup language, namely the XML standard published by the W3C organization. As explained above and by Dr. Almeroth, Arista's construction is inconsistent with how one of skill would understand the term. (Almeroth Decl. at ¶¶ 39-43). But more importantly, nothing in the intrinsic record suggests that the claims should be limited in the way that Arista suggests. Just the opposite, both the claims and the specification explicitly disclose that the invention should not be limited to any particular extensible markup language. First, the claims recite "***an*** extensible markup language (XML) format," which implies that there are multiple extensible markup language formats that fall within the scope of the claims. More explicitly, the specification repeatedly states that the invention is not limited to the particular XML language used as an example in the specification. (*See, e.g.*, '886 Pat., Ex. 2 at 3:26-29; 3:50-52; 4:27-30 (explaining that the invention is not limited to XML); *see also* Almeroth Decl. at ¶ 45). It is well-established that claims are not limited to the embodiments disclosed in the specification. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004). Arista's attempt to limit the claims to one embodiment should be rejected, because it is inconsistent with the law and with explicit and repeated teachings in the specification that the invention is not so limited.

B. "command line interface (CLI) parser" / "receiving, with a command line interface (CLI) parser" in Claims 1-5

Cisco's Construction	Arista's Construction
Claim term: "command line	Claim term: "receiving, with a command line interface

interface (CLI) parser" Proposed construction: a component of the routing system for analyzing command line interface (CLI) commands using a grammar	(CLI) parser" Proposed construction: "receiving": taking as an input "a command line interface (CLI) parser": a program that breaks down the individual sub-parts of a command using a CLI grammar

The parties first disagree with respect to the claim term being construed. Cisco proposes to construe "command line interface (CLI) parser," while Arista proposes to construe the broader phrase "receiving, with a command line interface (CLI) parser." Cisco's proposal should be adopted. The only difference between the proposals is Arista's proposal includes "receiving." But "receiving" is a common English word that is well understood by lay persons, and the claims are not using "receiving" in any specialized manner. Furthermore, Arista's proposed construction—"taking as an input" —does not offer any additional clarity to the word. *Duraflame, Inc. v. Hearthmark, LLC*, No. CV 12-01205 RS, 2013 WL 594241, at *9 (N.D. Cal. Feb. 14, 2013)(rejecting proposed construction in part because it "add[ed] nothing" to the claim being construed.). Therefore, construction of "receiving" is not necessary and the Court should only construe "command line interface (CLI) parser."

With respect to "command line interface (CLI) parser," the parties' competing constructions share certain similarities. But there are two main differences. First, Cisco construes the parser as "a component of the routing system," while Arista construes it as "a program." Cisco's proposed construction should be adopted because it is consistent with the intrinsic evidence. As shown in Fig. 1 and described in the accompanying text, the IOS/CLI parser 110 is a component of the routing system 100. ('886 Pat., Ex. 2 at 1:56-59; 3:10-22 ("***The components of routing system 100*** cooperatively operate to route or otherwise distribute signals received, . . . such command statements are received by routing system 100 through programming port 103, and are passed to ***IOS/CLI Parser 110***." (emphasis added))).

Second, Arista's proposed construction requires the parser to "break down the individual sub-parts of a command," while Cisco's proposed construction only requires that the parser "analyz[es] command line interface (CLI) commands." First, "individual sub-parts of a

command" is not a claim limitation. Arista's construction thus is an attempt to add a limitation to the claims and should be rejected for this reason alone. Second, parsing, as it is commonly understood in the art, involves an analysis of the input using a grammar. Parsing may involve breaking down the input into sub-components, but it is not required. (*See, e.g.,* Dictionary Of Computer Science, Engineering, and Technology (2001), Ex. 7 at CSI-CLI-00403897; McGraw-Hill Dictionary of Scientific and Technical Terms, 6th Ed (2003), Ex. 6 at CSI-CLI-00403883). Likewise, in the intrinsic evidence, "parsing" was defined in the context of U.S. Patent No. 5,778,223 as "the process of verifying syntactic correctness of a DDM string (DDM stream), and of translating it into a recognizable internal format," which does not require breaking down the DDM string into sub-parts ('233 Pat., Ex. 4 at 3:32-35).[4] Accordingly, Cisco's proposed construction better reflects the ordinary and customary meaning of "parsing."

C. "internetwork operating system (IOS) command line interface (CLI) parser subsystem" in Claims 2-4, 7-9

Cisco's Construction	Arista's Construction
a subcomponent of the internetwork operating system (IOS) for analyzing command line interface (CLI) commands using a grammar.	"internetwork operating system (IOS) command line interface (CLI)": the CLI interface designated by Cisco as IOS CLI and that is used for configuring, monitoring, and maintaining Cisco devices. Also referred to as 'IOS/CLI.' the entire phrase: processor or portion thereof that executes a program that breaks down the individual sub-parts of a command using the grammar of IOS/CLI

Cisco's proposed construction for "internetwork operating system (IOS) command line interface (CLI) parser subsystem" mirrors its proposed construction for "command line interface (CLI) parser," taking into account minor differences in the claim language. This is entirely consistent with the specification, which uses the term "IOS/CLI parser" and "IOS CLI parser subsystem" interchangeably. (*See, e.g.,* '886 Pat., Ex. 2 at 3:60-4:5 (explaining that Fig. 2 is a "flowchart 200 of a method for receiving and translating data using ***an IOS CLI parser***

[4] Prior art references listed on the face of the '886 Patent, such as the '223 patent, are part of the intrinsic record. *V-Formation, Inc. v. Bennetton Group SpA*, 401 F.3d 1307, 1311 (Fed. Cir. 2005) (citation omitted).

subsystem" but that at step 210 of flowchart 200, "input data is received at ***the IOS/CLI Parser 110*** of a routing system 100") (emphasis added)). It is clear from the specification that "IOS CLI parser subsystem" and "IOS/CLI Parser" refer to the same component of the routing system. Therefore, the Court should adopt a construction for "internetwork operating system (IOS) command line interface (CLI) parser subsystem" that mirrors the construction for "command line interface (CLI) parser," as Cisco proposes.

As explained above, it is contrary to well-established law to attempt to limit the claims to an embodiment in the patent. But Arista goes even further than that, and proposes that this claim term should be limited to products made by Cisco. There is nothing in the intrinsic or extrinsic evidence that even remotely indicates that the term should be so limited. Nothing in the specification indicates that "internetwork operating system (IOS) command line interface (CLI)" should be limited to Cisco-designed or Cisco-produced products. And indeed, Arista's construction is also contrary to the file history, where the Examiner rejected claim 2 (then pending claim 3) over non-Cisco prior art. (*See, e.g.,* May 5, 2008 Office Action, Ex. 3 at CSI-CLI-00024544 (rejecting then pending claim 3 in light of prior art assigned to Juniper Networks and to HP)). The patentee also did not argue that the claim was distinguishable over non-Cisco prior art because it was limited to Cisco-produced products. (*See, e.g.,* Sept. 5, 2008 Amdt., Ex. 3). Arista's proposed construction creates the absurd result that the only entity that can possibly infringe a claim in a Cisco-owned patent is Cisco itself. Such absurd results are not allowed and Arista's proposed construction should be rejected.

D. "XML tag" in Claims 1-10

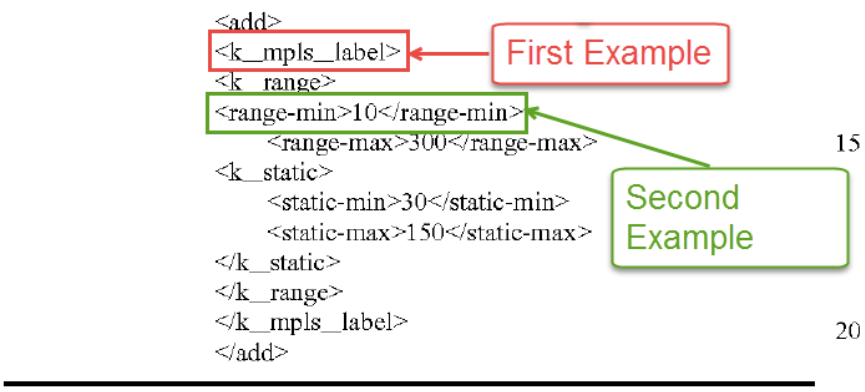
Cisco's Construction	Arista's Construction
one or a pair of XML indicators identifying data "extensible markup language" or "XML" as construed above	indicator that marks the start or end of an XML element and that is designated by a starting angle bracket, a corresponding closing angle bracket, and the content in between

The difference between the parties' proposed constructions for "XML tag" is simple. Cisco's proposed construction encompasses all embodiments of "XML tag" as disclosed in the specification, while Arista's proposed construction narrowly covers only one example disclosed in

the specification while excluding others. It is axiomatic that "a claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct." *Accent Packaging, Inc. v. Leggett & Platt, Inc.*, 707 F.3d 1318, 1326 (Fed. Cir. 2013) (citation omitted). For this reason, Cisco's prosed construction should be adopted while Arista's should be rejected.

The claims of the '886 Patent recite "at least one XML tag that includes an XML parameter to indicate the XML tag includes one or more CLI keywords." The specification shows several examples of XML tags that include one or more CLI keywords. The first example is by placing the CLI keywords within two angle brackets, such as "<k_mpls_label>" as shown in Table 1 in column 5 of the '886 Patent. The second example is for CLI keywords that are values, where the CLI keyword is placed between a pair of angle-bracketed indicators, such as "<range-min>10<range-min>" also shown in the same Table 1:

TABLE 1



('886 Pat., Ex. 2 at 5:10-21 (annotation added)). Importantly, the specification describes the second example as a ***single*** XML tag:

> In step 370 of flowchart 300, in one embodiment, each XML tag has the following rule applied to it. ***If an XML keyword tag is a parameter node tag***, the values inside the tag are extracted.

(*Id.* at 5:49-52 (emphasis added)). Thus, as it is described in the specification, an "XML tag" may be a single angle-bracketed indicator or a pair of indicators that includes CLI keywords. This is precisely Cisco's proposed construction. Arista's construction, on the other hand, would be limited to the first example and entirely excludes the second example. This is contrary to law and thus Arista's construction should be rejected. *Accent Packaging*, 707 F.3d at 1326.

E. "XML parameter" in Claims 1-10

Cisco's Construction	Arista's Construction
Plain and ordinary meaning, except for "XML" as construed above	indicator within an XML tag that signals that the tag includes a CLI keyword

The asserted claims recite "at least one XML tag that includes an XML parameter to indicate the XML tag includes one or more CLI keywords." In the context of the claim language and specification, and under the "heavy presumption" that claim terms retain their plain and ordinary meanings as understood by persons skilled in the art, *Elbex Video*, 508 F.3d at 1371, "XML parameter" has a plain and ordinary meaning that does not require further construction.

Arista's construction, on the other hand, does not offer any additional clarity on what an "XML parameter" is but rather merely repeats the claim language in the broader claim term. Substituting Arista's construction into the broader claim term, the limitation becomes "at least one XML tag that includes an 'indicator within an XML tag that signals that the tag includes a CLI keyword' to indicate the XML tag includes one or more CLI keywords." Arista's construction in the broader claim term repetitively recites that the XML tag includes indicators to indicate that the XML tag includes one or more CLI keywords. This shows that Arista's proposed construction is redundant and unnecessary. *See Merck & Co., Inc. v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) (rejecting construction that "renders other parts of the claim superfluous"); *see also Takeda Pharm. Co. v. Mylan Inc.*, No. 13-CV-04001-LHK, 2014 WL 5862134, at *23 (N.D. Cal. Nov. 11, 2014) (refusing to add language that would be "would be redundant.").

F. "parsing the output message to identify at least one CLI token" in Claims 1-10

Cisco's Construction	Arista's Construction
analyzing the output message to extract at least one unit of CLI characters in a sequence	breaking down the output message into its constituent character strings and determining that at least one such string corresponds to a CLI keyword or parameter

As is the case with other terms, the dispute here boils down the whether the Court should adopt Cisco's construction that encompasses all embodiments disclosed in the specification, or if the Court should adopt Arista's construction that reads out a preferred embodiment. Because

Arista's construction is contrary to law, Cisco's construction should be adopted. *Accent Packaging*, 707 F.3d at 1326.

Cisco's proposed construction is consistent with the plain and ordinary meanings of "parsing" and "token."[5] As explained above, Arista's construction of "parsing" as "breaking down" is improperly narrow. In addition, Arista also improperly attempts to limit "CLI token" to mean a constituent character string that corresponds to a CLI keyword or parameter. First, the term "constituent character string" is found nowhere in the intrinsic evidence, and Arista has cited to no extrinsic evidence to support its construction. (Dkt. 70-2, Ex. B at 22-23). Second, Arista offers a narrow construction that each CLI token must correspond to a single CLI keyword or parameter. Arista's construction is contrary to how one of ordinary skill in the art at the time of invention would understand the term "token." (Almeroth Decl. at ¶ 48). Such a construction is also contrary to the disclosures in the specification, where a CLI token may correspond to multiple CLI keywords. Specifically, the specification discloses that in an exemplary method of translating CLI statements into XML statements, a single CLI token may correspond to multiple CLI keywords when multiple CLI keywords are merged for a single CLI token. ('886 Pat., Ex. 2 at 6:30-40; *see also* Almeroth Decl. at ¶ 48). Therefore, Arista's construction would exclude an embodiment explicitly disclosed in the specification and thus should be rejected.

G. "wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands" in Claims 1-10

Cisco's Construction	Arista's Construction
Plain and ordinary meaning, except for "extensible markup language (XML) format" as construed	"an extensible markup language (XML) format": a format that complies with one of the versions of the XML standard published by the W3C organization "keyword": word describing an action or operation that the computer can recognize and execute

[5] *See, e.g.,* Dictionary of Computer Science, Engineering, and Technology (2001), Ex. 7 at CSI-CLI-00403897 ("Parsing: the process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar."); McGraw-Hill Dictionary of Scientific and Technical Terms, 6th Ed (2003), Ex. 6 at CSI-CLI-00403884 ("Token: 1. A distinguishable unit in a sequence of characters.").

above	The entire phrase: wherein the input command is written in an extensible markup language (XML) format, such that one or more CLI keywords, on the one hand, and the CLI parameters, on the other, are contained within respective XML tags, and the sequence of the tags complies with the sequencing rules for keywords and parameters in the CLI grammar and syntax

The claim limitation "wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands" is a detailed limitation that sufficiently describes what is claimed. Therefore, no further construction is necessary. Any additional construction of this long phrase likely would be cumbersome and confusing to the jury. *Toshiba Corp. v. Hynix Semiconductor, Inc.*, No. C–04–4708 VRW, 2006 WL 2432288, at *10 (N.D. Cal. Aug. 21, 2006) (rejecting "cumbersome construction" that would not "assist a jury [to] perform its duties").

As explained above in Section V.A, Arista's construction for "extensible markup language (XML)" is an improper attempt to limit the claims to an embodiment in the specification. Likewise, Arista's proposed construction for the broader phrase is a further attempt to limit the claims to embodiments, contrary to the law and the intrinsic evidence. In fact, Arista's proposed construction for the entire phrase is a blatant attempt to rewrite the claims so that it is limited to the embodiment shown in Fig. 3 and described at column 4:30 – 5:65 of the specification. This construction is contrary to well-established law. *Phillips*, 415 F.3d at 1313 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments."). Furthermore, it is also contrary to the specification, which explicitly teaches that the invention is not limited to this embodiment. ('886 Pat., Ex. 2 at 4:50-61 (stating that the steps of Fig. 3 are exemplary)). Accordingly, Arista's proposed construction should be rejected.

VI. CONCLUSION

For all the reasons stated above, Cisco's proposed constructions should be adopted and Arista's proposed constructions should be rejected.

DATED: November 2, 2015

Respectfully submitted,

/s/ Sean S. Pak

Kathleen Sullivan (SBN 242261)
kathleensullivan@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100

Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
John M. Neukom (SBN 275887)
johnneukom@quinnemanuel.com.
QUINN EMANUEL URQUHART & SULLIVAN LLP
50 California Street, 22nd Floor
San Francisco, CA 94111
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

Mark Tung (SBN 245782)
marktung@quinnemanuel.com
QUINN EMANUEL URQUHART & SULLIVAN LLP
555 Twin Dolphin Drive, 5th Floor
Redwood Shores, CA 94065
Telephone: (650) 801-5000
Facsimile: (650) 801-5100

Steven Cherny (admitted *pro hac vice*)
steven.cherny@kirkland.com
KIRKLAND & ELLIS LLP
601 Lexington Avenue
New York, New York 10022
Telephone: (212) 446-4800
Facsimile: (212) 446-4900

Adam R. Alper (SBN 196834)
adam.alper@kirkland.com
KIRKLAND & ELLIS LLP
555 California Street
San Francisco, California 94104
Telephone: (415) 439-1400
Facsimile: (415) 439-1500

Michael W. De Vries (SBN 211001)
michael.devries@kirkland.com
KIRKLAND & ELLIS LLP

333 South Hope Street
Los Angeles, California 90071
Telephone: (213) 680-8400
Facsimile: (213) 680-8500

Attorneys for Plaintiff Cisco Systems, Inc.

PROOF OF SERVICE

I, Jason L. Liu, declare that I am over the age of eighteen (18) and not a party to the entitled action. I am an attorney in the law firm of Quinn Emanuel Urquhart & Sullivan LLP, and my office is located at 50 California Street, 22nd Floor, San Francisco, CA 94111. On November 2, 2015, I caused to be filed the following:

1) PLAINTIFF CISCO SYSTEMS, INC.'S OPENING CLAIM CONSTRUCTION BRIEF

2) DECLARATION OF KEVIN C. ALMEROTH SUBMITTED IN SUPPORT OF PLAINTIFF CISCO SYSTEMS, INC.'S OPENING CLAIM CONSTRUCTION BRIEF, INCLUDING APPENDICES A-E

3) DECLARATION OF KENNETH SUH IN SUPPORT OF PLAINTIFF CISCO SYSTEMS, INC.'S OPENING CLAIM CONSTRUCTION BRIEF, INCLUDING EXHIBITS 1-8

with the Clerk of the Court using the Official Court Electronic Document Filing System, which served copies on all interested parties registered for electronic filing.

I declare under penalty of perjury that the foregoing is true and correct. Executed on November 2, 2015, at San Francisco, California.

/s/ Jason L. Liu
Jason L. Liu